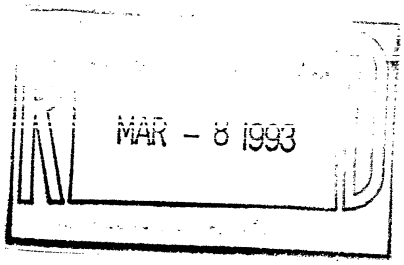
cc: Randall, me
file: Lavitsky contract folder

Lavitsky Computer Laboratories, Inc.

PO Box 7446, Somerset, NJ 08875 908-560-0114



Jeff Porter
Commodore Business Machines
1200 Wilson Dr.
West Chester, PA 19380

March 1, 1993

Dear Jeff,

As you suggested, I am including a brief synopsis of the work we performed over the last billing period (February).

- ASCO 2300 Device Driver
- Future directions:
  - Amiga Hardware Design/Status Review
  - Amiga Software Design/Status Review
  - VCOS Internals Review
    - Multitasking issues
    - VRM API/Internals
    - Changes to VE
- VCAS Work:
  - Removed all references to malloc/free, now uses AllocVec/FreeVec (will switch to AllocVecPooled/FreeVecPooled).
  - Re-compiled all software under SAS/C 6.2
- VCSIM:
  - Began porting analysis
- Tools Work:
  - Ported 1.3.1 d32sim in preparation for VCSIM
    - noted performance improvement over version compiled with older SAS/C
    - began recompiling all tools under SAS/C 6.2
    - began compiling list of problems with source and porting for John Lynch.
- VCOS SDK Documentation Review
- MPEG Audio Format Decoder

I am also including copies of some memos that I generated for the DSP group at Murray Hill. Please forward copies to Randell. I am pleased with the direction we have taken with the hardware and I look forward to it's completion.

Sincerely,

Eric Lavitsky
President
Lavitsky Computer Laboratories, Inc.

Lavitsky Computer Laboratories, Inc.

PO Box 7446, Somerset, NJ 08875 908-560-0114

From: Eric Lavitsky 2/18/93
To: DSP Group

Re: Perceived problems and issues in VCOS and related software tools.

There are a number of problems and issues in the VCOS software that I would like to bring to your attention. Commodore needs to have these resolved prior to their inital customer release. These problems are categorized by subsystem and listed in order of severity:

VCAS (Release 1p02):

1. VCAS does not lock global data sections when accessing them. This is a potentially fatal condition on a preemptive shared-memory multitasking system such as ours and will also be a problem under Unix, Windows NT, etc. If a task is preempted while updating critical global data in VCAS, the integrity of the entire system will be compromised. The solution is to identify all areas where global data is being accessed, attempt to minimize the duration of these accesses to the finest granularity, and provide a mechanism to lock the global data during access. I propose two new VE functions:

```
veLock(void *semaphore)
veUnLock(void *sempahore)
```

These functions shall be wrapped around all access to global data sections. Multiple semaphores may be used to access smaller sections of data. This will permit greater system throughput. The actual implementation may use only one global semaphore or even simply prohibit task rescheduling, but this is, of course, left up to the individual system designer.

2. VCAS does not free all memory when it exits. On our system, it is possible for VCAS to be flushed from the system when there is no application using the DSP. VCAS currently loses between 300 and 400 bytes on exit. Also, if VCAS encounters an error condition on startup (such as not being able to open the OS files), it does not free all it's resources properly (close all files, etc.).

3. VCAS does not account for systems with large caches that lack bus snooping hardware for cache management. Care must be taken when accessing data on the host that has been or will be modified by the DSP. There are two software approaches to solving this problem:

1. Use an MMU to tag all memory allocations related to the DSP as non-cachable.

2. Explicitly flush the system caches as required.

We can implement solution 1 for our initial release on our 68030 and 68040 systems by making all veAlloc... calls and all VCAS internal allocation allocate non-cachable memory. We cannot implement this solution for our systems which do not have an MMU. Specifically, all of our Amiga 1200 models (which are currently selling over 50K units per month) use a 68EC020 which does not have an MMU. There are also versions of the 68030 and 68040 which third party accellerators and possibly systems that Commodore will release in the future use which do not

incorporate an MMU (68EC030, 68EC040). Any VCAS calls which reference data which may be modified by the DSP should explicitly flush the system caches. I propose the following VE function to implement this:

```
veFlushCache(void *ptr, unsigned long size);
```

VCD (Release 7.22.92):

1. VCD does not vector through the routine vc_Exit for all fatal error conditions. Instead, it sometimes calls exit() directly. This prevents properly cleanup after a fatal condition in VCD. This problem can easily be remedied by replacing all references to exit() with vc_Exit().

2. VCD busy waits when waiting for user input. It also calls check_bp even if there are no DSPs running. The routine no_key_hit should be modified to check if there are any DSPs running, or any iostreams active, then call check_bp only for those active. Otherwise, it should loop for user input, or sleep if possible. This will reduce overhead during idle activity and make it easier for developers to debug applications in a multitasking environment.

Tools (Release 1.3.1):

The software tools contain problems too numerous to list. The source code generates thousands of compiler warnings on our system, many of which are potentially serious. It should take minimal effort to clean up the code using ANSI standard function prototypes at the very least. This would expose both obvious and subtle errors in argument passing, type casting etc. Most modern compilers provide tools for bringing source up to date with ANSI standards, including automatic generation of function prototypes.

I would be happy to discuss any or all of these issues in further detail. I look forward to their resolution before the first release to the Amiga developer and user communities.

Sincerely,

Eric Lavitsky

Lavitsky Computer Laboratories, Inc.

PO Box 7446, Somerset, NJ 08875 908-560-0114

From: Eric Lavitsky 2/24/93
To: DSP Group

Re: Considerations for architectural and internal changes in the VCOS product rewrite.

In addition to my previous memo, there are a number of items that are very important to consider for the rewrite of VCOS. Addressing these issues will improve the portability and extensibility of the VC/VE/VCD code.

VE:

• The "dspid" is not passed as an argument to the functions veLogToDspPhys or veDspPhysToLog. This presents a problem on systems where a mix of bus-master and non-bus-master DSP's exist or even multiple non-bus-master cards where the software architecture provides a centralized mechanism for dereferencing VE calls.

VCAS:

• "int" is used too liberally for arguments and returns. It is not clear what "int" really means on all architectures. For instance, on an Amiga, it may be either 16 or 32 bits depending on what choice the programmer makes, but it defaults to 32 bits, whereas most PC compilers make it 16 bits by default. ints are also limiting for future expansion.

• Tags or some type of tag based argument passing should be supported for VCAS functions.

• Address pointers are often declared as "char *". It is incorrect to use signed values for address calculation. All pointers should be either "void *" or there should be typedefs for address pointers etc..

• Error messages are currently output using multiple calls to vePrintf. This results in a cosmetically unattractive presentation to the user under both Windows and AmigaDOS, since multiple dialog boxes must be opened to relay the message. All error messages should be output with one call to vePrintf. All formatting should be handled by the vePrintf function.

General:

• Error strings should be separated out into their own file. Some kind of function should be provided for returning an error string given an error id. This will make it easier to localize VCOS messages for other countries.

• No version or author/copyright information is maintained in VE or any of the MML's. It would be nice to have a VC function to determine the version/id string of an MML. Having a version number for VE would make it possible to implement new features in VCAS without forcing users to replace VE. Version checking would be done within VCAS before

attempting new VE calls. This will further insulate the VCAS developers from hardware vendors who may not be able to react quickly to a new release.

• The preprocessor defines which VC/VCD set: HUGE GLOBAL BITSET SIGN OVERFLOW, may conflict with some other systems headers (they do on the Amiga). A more specific naming scheme should be used for defining such symbols, e.g.: VC_HUGE, VC_GLOBAL, VC_BITSET, VC_SIGN, VC_OVERFLOW.

• Configuration information is stored in files. There are, however, no defined external software interfaces for communicating configuration information. If data structures and routines were defined and documented for storing and communicating this information, then systems which can autoconfigure certain hardware subsystems could (now, or in the future) eliminate the need for having configuration files. This would be desirable since files are fragile and can be "damaged" by a casual user.

• While having TBD and LDF files in ASCII form is convenient for the application programmer during development, it may not be so for a shrink-wrapped product. Again, casual users should not be able to edit vital configuration information if the application developer did not intend them to. Some (combined) binary file format should be adopted for TBD/LDF etc. files. It would also be nice to have function calls or data structures which make it possible to build a task dynamically.

• And finally a few words on project management. Source code control is somewhat in place on dsps, but not everything is available yet, including all the MML's built properly for big and little endian architectures, the kernel, etc. A well supported and maintained system of reporting and tracking bugs and feature enhancements should be maintained. A database that is accessable to everyone in the group should be put in place for this purpose. Using dsps as a base for this would be appropriate. It should be backed up by a newsgroup where all reports and comments can be shared in an open (to the group only) electronic forum.

I look forward to seeing these issues addressed and would be happy to discuss any of them in further detail.

Sincerely,

Eric Lavitsky

Lavitsky Computer Laboratories, Inc.

PO Box 7446, Somerset, NJ 08875 908-560-0114

From: Eric Lavitsky 3/2/93
To: DSP Group

Re: Problems and issues related to the DSP3210 Software Tools.

As a representative of a major VCOS and DSP3210 OEM which has been involved in the porting of the AT&T VCOS and DSP3210 Software tools, we have had extensive experience with various releases of these software systems. In our experience with the tools, we have encountered barriers to what should be a fairly straightforward process:

- We are not notified of new releases of the software. As a customer, we do not want to have to chase down new releases, nor do we want to "discover" bugs that were solved elsewhere long ago. The porting center at Murray Hill has been a reliable source of contact and software support for our organization. It makes sense for us to have the tools managed at the porting center. This way we have one point of contact and one central location for software support and porting.

- There are numerous problems with the tools source code:

    - Proper ANSI function prototypes are not used or incorrect prototypes are given.
    - Variables are declared and then never referenced.
    - Functions are declared without a return type (not even void).
    - Functions and data structures are cast to the wrong type.
    - Local variables are used before they are assigned a value.
    - Some assignments use the incorrect value or define (e.g. INT instead of T_INT)
    - Expected arguments are occasionally not passed to functions.
    - Structures or functions are defined without checking for existing definitions.
    - Some definitions conflict with ANSI standard functions or structures.
    - Header files and methods are not standardized.
    - System dependencies are not organized in any central location.
    - No validation suite is provided for OEM's who are porting the tools.

Every tool exhibits some problem, and has since at least release 1.1, even though we have brought many of these problems to the attention of the tools group from time to time. As an example, we recently ported and compiled d32sim. This code alone produces well over 300 error messages on our C compiler, some of which had to be coded around to work properly. Most, if not all of the other tools exhibit similiar characteristics.

We believe that these problems should be addressed immediately. The tools are an important part of the DSP3210 and VCOS solution and, in our opinion, critical to its success on our platform.

Sincerely,

Eric Lavitsky
President
Lavitsky Computer Laboratories, Inc.